Problem 1: Least Absolute Deviation (15 points)

In class, we have assumed the following data generative model

$$y = f(\mathbf{x}) + \epsilon$$

where ϵ follows a standard gaussian distribution, i.e., $\epsilon \sim \mathcal{N}(\epsilon|0,1)$. Assume a linear model for $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$ (the bias term is ignored here). We now modify the data generative model by assuming that ϵ follows a Laplacian distribution whose probability density function is

$$p(\epsilon) = \frac{\lambda}{2}\exp(-\lambda|\epsilon|)$$

where λ is a positive constant. For more about Laplacian distribution please check the following wiki page `http://en.wikipedia.org/wiki/Laplace_distribution`.

Based on the above noise model about ϵ, derive the log-likelihood for the observed training data $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ and the objective function for computing the solution $\mathbf{w}$. Does the problem have a closed form solution like Least Square Regression?

1. Derive the log-likelihood (L):

   Laplacian: $p(\epsilon) = \frac{\lambda}{2}\exp(-\lambda|\epsilon|)$

   $$\mathrm{L} = \prod_{i=1}^{n} \frac{\lambda}{2}\exp\left(-\lambda\left|y_i - w^\top x_i\right|\right)$$

   $$\log \mathrm{L} = \sum_{i=1}^{n} \log\left(\frac{\lambda}{2}\right) - \lambda\left|y_i - w^\top x_i\right|$$

2. Objective Function:

   $$\min_{w} \sum_{i=1}^{n}\left|y_i - w^\top x_i\right|$$

3. This problem does not have a closed form solution (like the Least Square Regression) because the use of the absolute value function makes it piecewise.

Problem 2: Fitting Polynomial Functions (35 Points)

In class, we see that, a high-order polynomial, despite exhibiting superior fit to the training data, could result in diminished accuracy when applied to unseen testing data. Here, we try to observe it and try to improve it. Code to generate data and fit a model using sklearn is given in "p2.py" file. We measure the performance by the root mean square error (RMSE[1])

Each data point i originally has a feature $\mathbf{x}_i \in \mathbb{R}$, and a target $\mathbf{y}_i$. We first transfer it to polynomial features as $\phi(\mathbf{x}_i) = (\mathbf{x}_i, \mathbf{x}_i^2, \mathbf{x}_i^3, ..., \mathbf{x}_i^d)^\top$, where d is the degree. With a linear model $\mathbf{w} \in \mathbb{R}^d$ and w_0 to be a intercept term, the prediction is given by $\phi(\mathbf{x}_i)^\top \mathbf{w} + w_0$. Let $\Phi = (\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), ..., \phi(\mathbf{x}_n))^\top$ be a $\mathbb{R}^{n\times d}$ matrix, i.e., each row of Φ representing the polynomial features of one data point. We consider the following problem

$$\min_{\mathbf{w}} \frac{1}{2}\|\Phi\mathbf{w} + w_0 - \mathbf{y}\|_2^2 + \frac{\lambda}{2}\|\mathbf{w}\|_2^2,$$

1) Fix $n\_train = 10$ and $\lambda = 0$, vary degree in $[1, 5, 10, 20]$, i) Compare how training RMSE change. ii) Compare how testing RMSE change. Write your response to these questions in pdf (same below).

2) Fix $\lambda = 0$. Increase n_train to be 10000. Vary degree in $[1, 5, 10, 20]$. How are the training RMSE and testing RMSE compared to 1)?

3) Let $n\_train = 10$, $\lambda = 0.001$. Vary degree in $[1, 5, 10, 20]$. How are the training RMSE and testing RMSE compared to 1)?

1. Fix $n\_train = 10$ and $\lambda = 0$, vary degree in $[1, 5, 10, 20]$

Degree	1	5	10	20
Training RMSE:	0.7136	0.0894	0.0915	0.0786
Testing RMSE:	0.8619	0.4864	1.7818	7.0587

Degree	1	5	10	20
Training Analysis	under fit	good fit	good fit	very good fit
Testing Analysis	under fit	good fit	over fit	very over fit

2. Fix $n\_train = 10000$ and $\lambda = 0$, vary degree in $[1, 5, 10, 20]$

Degree	1	5	10	20
Training RMSE:	0.7246	0.3353	0.2999	0.2945
Testing RMSE:	0.8282	0.5266	0.5060	0.5043

Degree	1	5	10	20
Training Analysis	under fit	decent fit	decent fit	decent fit
Testing Analysis	under fit	good fit	good fit	good fit

3. Fix $n\_train = 10$ and $\lambda = 0.001$, vary degree in $[1, 5, 10, 20]$

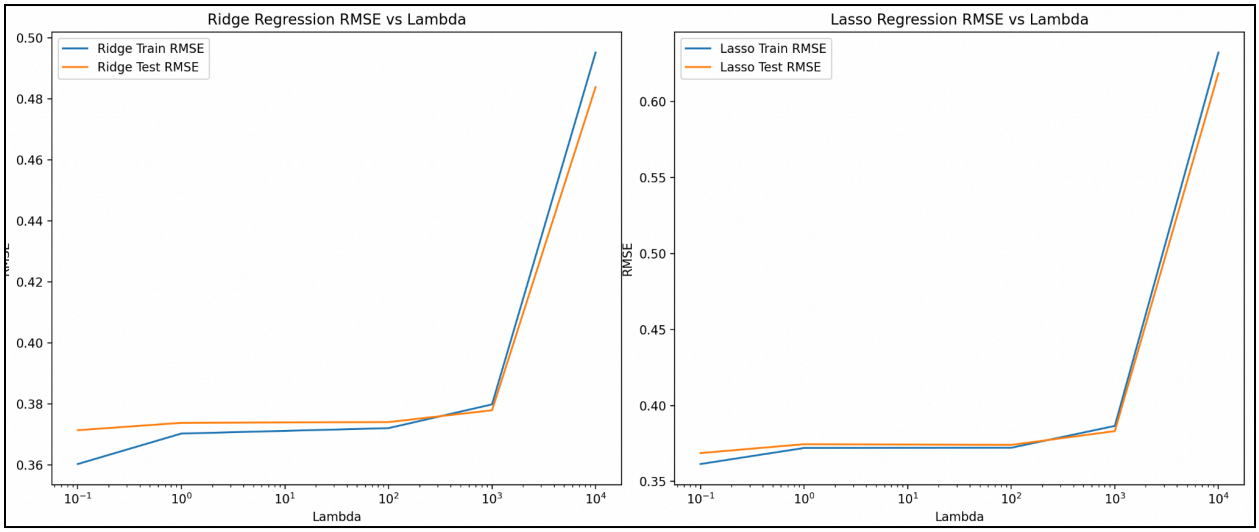
Degree	1	5	10	20
Training RMSE:	0.7136	0.3983	0.3379	0.3343
Testing RMSE:	0.8619	0.9823	0.7419	1.6834

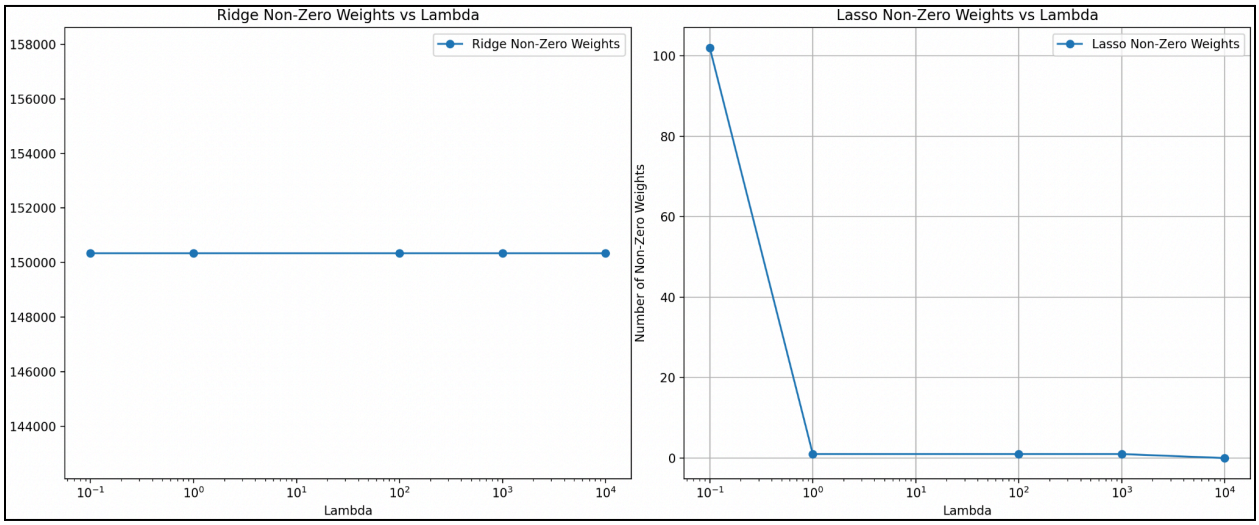
Degree	1	5	10	20
Training Analysis	under fit	decent fit	decent fit	decent fit
Testing Analysis	under fit	under fit	under fit	over fit

Problem 3: Ridge Regression and Lasso (50 points)

(1) $\lambda = 0.1$, Non-Zero Coefficients:
   (a) Ridge: 150348
   (b) Lasso: 102

(2) See Plots Below:





Observations:
As can be seen in both Ridge Regression and Lasso RMSE, the training and testing are at first far apart, then converge right before a large spike at Lambda = 1e4. This indicates an initial underfit, then a good fit right before an overfit at the spike.

Additionally, in the Non-Zero Weights graphs, one will note that Ridge Regression stays at the same number of non-zeros for all Lambda while the Lasso starts at 102 coefficients for Lambda = 0.1, then falls to 1 coefficient for Lambda = 1, 100, and 1000, before falling to 0 for Lambda = 1e4.

(3) Best λ and testing error:
Ridge: $\lambda = 0.1$, RMSE = 0.3714
Lasso: $\lambda = 0.1$, RMSE = 0.4021